COCPIT: Collaborative Activity Planning Software for Mars Perseverance Rover

Ivy Deliz
ASRC Federal
NASA Ames Research Center
Moffett Field, CA 94035
Ivy.Deliz@nasa.gov

Andrea Connell
Jet Propulsion Laboratory,
California Institute of Technology
Pasadena, CA 91109
Andrea.M.Connell@jpl.nasa.gov

Chet Joswig
Jet Propulsion Laboratory,
California Institute of Technology
Pasadena, CA 91109
Joseph.C.Joswig@jpl.nasa.gov

Jessica J. Marquez
NASA Ames Research Center
Moffett Field, CA 94035
Jessica.J.Marquez@nasa.gov

Bob Kanefsky
San José State University Research Foundation
NASA Ames Research Center
Moffett Field, CA 94035
Bob.Kanefsky@nasa.gov

***Abstract*—Since landing on the Martian surface, the Perseverance rover has relied on a distributed team to generate commands for exploring its new environment each sol (Martian day). The team uses a complex suite of software tools to accomplish this challenging task in time for the next window of opportunity to send commands to the rover. A key piece of this software ecosystem is COCPIT (Component-based Campaign Planning, Implementation, and Tactical). COCPIT is part of the next generation of planning and scheduling software tools developed by NASA's Jet Propulsion Laboratory in partnership with NASA's Ames Research Center.**

COCPIT is a web-based application that allows users to collaboratively view and update the Perseverance rover's activity plans, continuously verify that the plan satisfies constraints, assign targets for directing scientific instruments, document science intent, and model power and data resources. Mars Surface Operations requires diverse expertise from team members within the Engineering, Science, Robotic, and Instrument Operations groups, distributed across North America and Europe. In order to improve efficiency and reduce risk, all teams are able to review and edit their activities simultaneously and see the effects on the plan in its entirety.

As part of the Ground Data System (GDS) tool suite, COCPIT is responsible for the activity plan. It provides specialized views that allow operators to understand where there may be room for additional observations, see whether any planning constraints are being violated, and confirm that energy usage and data generation are within the defined limits. It contains details such as which filters a camera will use for a given observation, what the resolution of the images should be, where to store the data onboard, and how long the observation is expected to take. It predicts when specific data will be downlinked from the rover to a passing orbiter, so that the team knows when to expect that data on Earth for evaluation in future planning. Ultimately the information from the COCPIT plan is translated to sequences that will be bundled and radiated to Perseverance for execution. The COCPIT tool is used throughout all planning phases.

Table of Contents

1. Introduction

The Mars 2020 mission's Perseverance rover landed in Jezero Crater on February 18, 2021. The mission's surface operations team assesses the rover's state, examines its surroundings, and plans its daily activities. This requires diverse expertise from team members within the Engineering, Science, Robotic, and Instrument Operations groups, distributed across North America and Europe. The operations team uses a complex suite of software tools to accomplish this challenging task as

efficiently as possible. These tools are part of the Mars 2020 mission's Ground Data System (GDS) [1]. GDS is divided into subsystems, one of which is the Activity Planning and Sequencing Subsystem (APSS), responsible for activity planning, resource modeling and creating sequences to command the rover. As part of APSS, the COCPIT (Component-based Campaign Planning, Implementation, and Tactical) tool is responsible for the activity plan in the mission's software ecosystem.

In order to improve efficiency and reduce risk in comparison with previous Mars missions, it was decided that all team members should be able to simultaneously review and edit rover activities and see their effects on the shared plan. Additionally, the mission sought to improve processes by allowing external tools to interface with plans programmatically via APIs. Thus, COCPIT was developed – a web-based application that allows users to collaboratively view and update the Perseverance rover's activity plans, create and continuously verify constraints, assign targets for directing scientific instruments, document science intent, and model power and data resources.

The COCPIT activity plan contains the intricate details of what the rover should do on Mars, such as which filters a camera will use in a given observation, what the resolution of the images should be, how many photos are in a panorama, how far the rover should drive, where to store data onboard and how to prioritize sending it, how long an observation is expected to take, and many more details. It models when data products will be downlinked (transmitted over UHF radio) from the rover to a passing orbiter, so that the team knows when to expect that data on Earth for evaluation in planning future sols (Martian days). Ultimately, the information from the COCPIT plan is translated to sequences that will be bundled and radiated to Perseverance for execution.

COCPIT is part of the next generation of planning and scheduling software tools developed by NASA's Jet Propulsion Laboratory in partnership with NASA's Ames Research Center. COCPIT is built on top of existing functionality from Playbook, a web-based software tool built by the Scheduling and Planning Interfaces for Exploration (SPIFe) Team from the Human-Computer Interaction Group at NASA Ames Research Center. Playbook has been used to view and execute astronaut schedules [2], as well as study crew autonomy in NASA analogous missions [3]. Taking advantage of Playbook's existing collaborative features, timeline views and constraint and planning unit structures, COCPIT inherits functionality of the tool that already existed. Furthermore, COCPIT includes new features that are required for a robotic mission, in addition to adding interfaces to import and export data across the GDS system.

2. Architecture

COCPIT is a JavaScript application with a node.js server, and JavaScript front-end using HTML, CSS and Backbone.js for model-view organization and event handling, d3 and svg for timeline visualization and a redis database for storing transactions as the plan is built. Plan model synchronization between server and client is done via web sockets.

The COCPIT plan is persisted in an XML format, Rover Markup Language (RML), which also serves as the interchange format with other ground software and is schema-validated to ensure the proper structure of the file. The RML schema has been used for previous Mars missions, and has been extended to meet the needs of Mars 2020. On plan load, the data is imported from RML and served to browser clients who can then manipulate the plan via the COCPIT User Interface. All changes are sent to the server as messages and distributed to all clients viewing the same plan. Periodically, the RML file is exported and saved to a permanent cloud storage.

COCPIT servers are deployed in a Docker container to Amazon ECS (Elastic Container Service) Clusters via Terraform. All Mars 2020 GDS tools share a Single Sign-On layer for authentication, and an S3/Elasticsearch Data lake [1].

COCPIT is used collaboratively by an international team of scientists and engineers. In standard Mars surface missions, the operations team is co-located at the Jet Propulsion Laboratory for approximately the first 90 sols, and subsequently distributed from sponsoring institutions. On Mars 2020, prioritizing a first-class remote user experience turned out to be a critical benefit when the COVID-19 pandemic forced the majority of the team to be distributed – in many cases working from home – from the start of the mission.

3. Planning Phases

COCPIT supports all planning phases within the Mars 2020 mission. COCPIT's role is described in each planning phase below.

Parcel Development

Parcel development is the process in which the operations team defines and validates reusable Activities and Components. A separate tool called Activity Dictionary Editor is used to define and edit Activity types. Activities are the smallest planning units. An Activity type definition includes parameters and formulas, power and data effects, heating requirements, hardware state requirements, and more. Editing the Activity Dictionary creates a versioned copy, which can be imported into COCPIT and used to create a Component Library. A Component Library combines Activities from the Dictionary into higher level planning units. Each Component is a template that contains one or more Activity Groups, which in turn

contain one or more Activities. Activities in an Activity Group can either be serial (one starts after another ends) or parallel (start at the same time as each other). Activity Groups determine the precise order of Activities in the final plan and sequence, and this order cannot be changed outside of the Component Library. Components are logical groupings which allow for the convenience of putting multiple related Activity Groups into a plan together.

A Parcel includes an Activity Dictionary and Component Library among other assets. Once a Parcel is complete, it can be used to create a COCPIT plan. Components from the Component Library can be added to the plan, and their scheduling parameters can be adjusted as needed.

Strategic Planning

Strategic Planning models first-time activities or events that have not been executed on Mars before. A plan fragment is created in COCPIT to help the operations team visualize the required components and activities, set parameters as appropriate, consider timing constraints, and view heating requirements and resource impacts. The Strategic Planning process also helps identify additional validation that may be needed before the operation can be performed on Mars, and allows the team to prepare sequences in advance.

Plan fragments can also be developed for complex or recurring events, like sampling and caching. These serve as centralized reference points that can be updated based on process changes or lessons learned.

When a strategic plan fragment is finished, the planning units can be copied into the Look-Ahead Plan for Campaign Implementation.

Campaign Implementation

During Campaign Implementation, the operations team creates a Look-Ahead Plan in COCPIT to schedule a high-level plan (called a sol path) for the next 3-5 sols. It includes X band and UHF communication passes, which are pre-negotiated opportunities for the mission to send and receive information through the Deep Space Network and the Mars Relay Network of orbiting satellites. The Look-Ahead Plan also includes science and engineering activities that have been prioritized for the upcoming sols. The Look-Ahead Plan is used to ensure that the mission is progressing toward science goals and campaigns appropriately, and to provide an estimation of which types of activities will fit into the upcoming sols given the available data downlink volume, power, and UHF pass timing.

At the completion of each Campaign Implementation shift, one or more sols are "torn" from the Look-Ahead Plan and promoted to the next day's Tactical Uplink process.

Tactical Uplink

During Tactical Uplink, engineers and scientists use COCPIT to determine the details of what the Perseverance rover will be doing on Mars in the next planning cycle, which may span one or more sols. While the decisions made in Campaign Implementation provide a guide for Tactical Uplink, the plan often requires changes based on the latest data received from the rover. If the telemetry reveals anything unexpected, the team must respond to diagnose or resolve issues. Even when everything went perfectly, science targets need to be selected based on updated imagery and location information, engineering activities finalized, and drive paths determined. Team members update the activity plan to meet scientific and engineering goals, while confirming that resources and constraints are met. Sequences are generated for the planned activities, and flight rules are checked to ensure the safety of the rover.

In accordance with mission principles for increased operational efficiency [1], COCPIT has been designed and developed to enable a shorter Tactical Uplink process than previous missions. A more efficient uplink process means the team will have time to wait for more downlink telemetry before starting planning, allowing Perseverance to do more on each sol and reach mission goals more quickly.

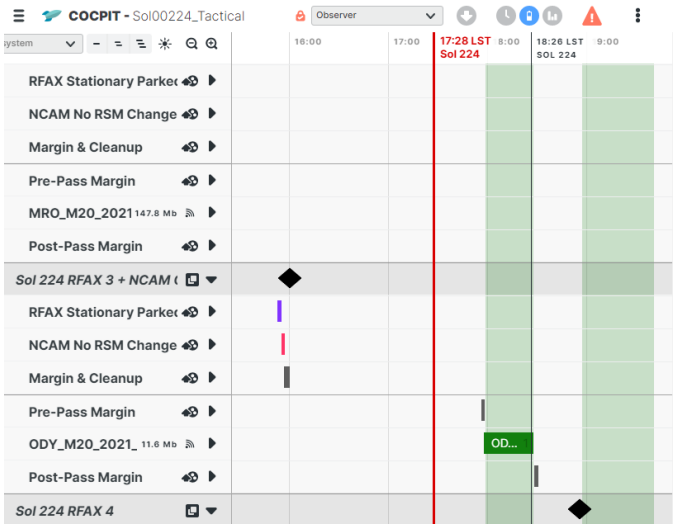


Figure 1. Red line showing current time on Mars

Tactical Downlink

While the Uplink Planning cycle is always looking into the future, the Tactical Downlink operators use COCPIT to view the currently executing sol's timeline to get context of the data being received, as well as to inform when new data is expected. COCPIT has a Marcus Bains vertical red line, as shown in Figure 1, that updates to show the current time on Mars. This is an example of a feature inherited from Playbook which is used for real-time execution.

The planning phases are cyclical and interconnected. Strategic

Planners continuously refine Parcel's Activity Dictionary and Component Library, and existing plans can be migrated to a newer Parcel as needed. Strategic plan fragments are copied into Campaign Implementation, and Campaign Implementation plans feed directly into Tactical Uplink. As new information is received from Tactical Downlink, both the Tactical Uplink and Campaign Implementation plans may need to be adjusted for future sols. COCPIT plays a key role in all of the phases and the development team continues to build new functionality and usability improvements to continue exploring Mars more efficiently.

4. User Interface

Navigator

The COCPIT Navigator is a companion view that allows creating, searching, browsing, and bookmarking plans, Component Libraries, and folders in COCPIT. The Navigator is inherited by COCPIT from Playbook but additional features have been added for supporting Mars 2020 mission workflows and the increasing number of items that need to be accessed. The Navigator serves as the entry point to top-level operations like renaming, moving, and duplicating a plan. It can also be used to migrate a plan to a different Component Library, migrate a Component Library to a different Activity Dictionary, update a plan's time span, and bookmark an item.

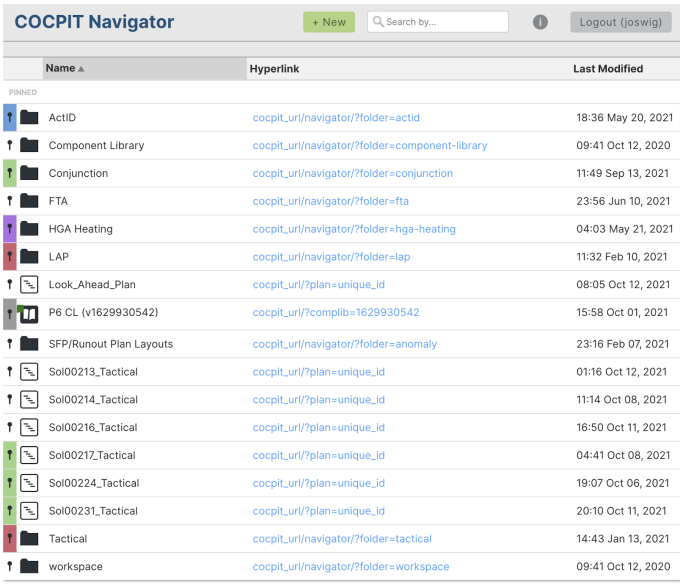


Figure 2. COCPIT Navigator Homepage

The home page of the Navigator, shown in Figure 2, only displays objects that have been bookmarked by users. These bookmarks are kept updated by user processes during the planning phases such that all important objects are easily accessible each day; users typically do not need to search to access the plan they need. For instance, the current Look-Ahead Plan and Tactical Plan are bookmarked, in addition to plans from several previous sols and the approved Component Library. Colors can be associated with the objects in the Navigator for visual tagging.

A workspace folder enables each user to have their own sandbox location to create and duplicate plans that may not be part of the regular planning phases.

To find objects that are not bookmarked, a search bar is provided where users can search by name or by using ElasticSearch query strings for more complex queries.

Double clicking on a folder, plan, or Component Library from the Navigator will open it in COCPIT.

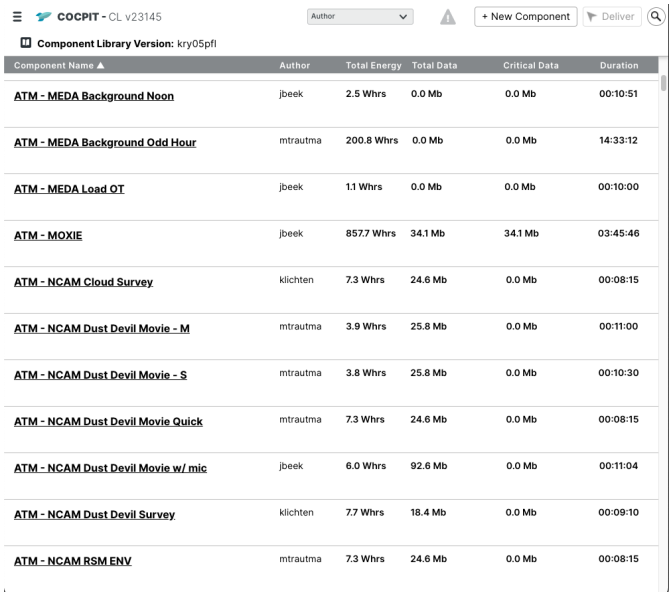


Figure 3. Component Library View

Component Library View

During Parcel Development, a Component Library is built and refined in COCPIT. The Component Library View, shown in Figure 3, is a table list of Components which shows the name, author and modeled information like Total Energy, Data Acquired and predicted duration.

Users can select one of the table items to open the Component Authoring View, shown in Figure 4. A Component is a structure that contains one or more Activity Groups, each of which contain one or more Activity instances

During Component Authoring, the number and order of the Activities in an Activity Group is defined, along with whether the activities should be executed serially or in parallel. Activity, Activity Group and Component parameters can be edited. If a Component has more than one Activity Group, dependency constraints can be added to control relative ordering between them. Time constraints can also be set to provide a time range for execution based on a specific time (e.g. 14:00) or geometric events (e.g. one hour after sunset). When the Component is eventually added to a plan, it will use

the Component Library configuration by default but can be refined and edited for the particular needs of that sol.

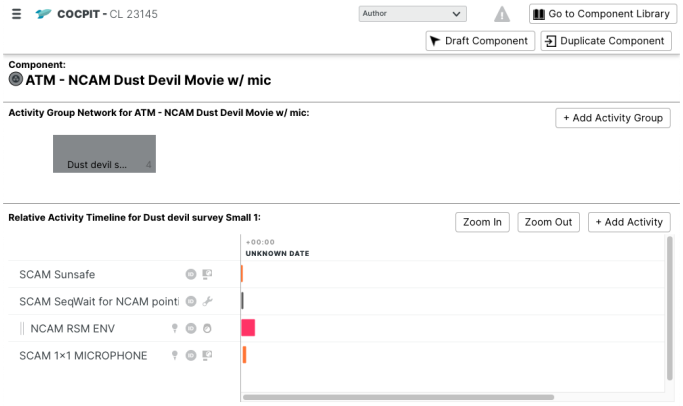


Figure 4. Component Authoring View

In a Tactical Plan, the Component Library View is accessible as a separate view showing all the Components in the Component Library. These Components are used for reference only and are not editable. Exceptions are made when necessary by creating a Tactical Component which copies a template from an existing instance of a component from the plan. The structure of a Tactical Component is completely editable until it is scheduled in the plan. When a Tactical Component is created, it only exists in that version of the plan. If the Tactical Component is deemed useful, the user can opt to "Draft" it to a Component Library for future inclusion in a Parcel.

Details Panel

Selecting a planning unit in any of the COCPIT views opens a Details Panel on the right side of the interface that displays all of the planning data for the selected planning unit (Figure 5). From the top, all planning units have the name and type, schedule information (i.e. start, end and duration), resource effects and parameters sections. Some planning units also have type-specific sections like constraints (for Activity Groups) and Science Intent (for Activities). Sections of the Details Panel can be collapsed when they are not in use to make scanning and scrolling through the panel easier.

Editing planning units is done mainly through the Details Panel. On selection, the panel comes up and displays widgets for editing parameters. Different parameter types have type-specific editing interfaces, for example drop downs for enumeration types, check boxes for booleans, or text inputs for strings or numbers. Fields are editable as a function of the user's current role and planning process phase (as described in Section 5). Additional information like data and power resource effects are not editable because they are derived from parameters and formulas defined in the Activity Dictionary.

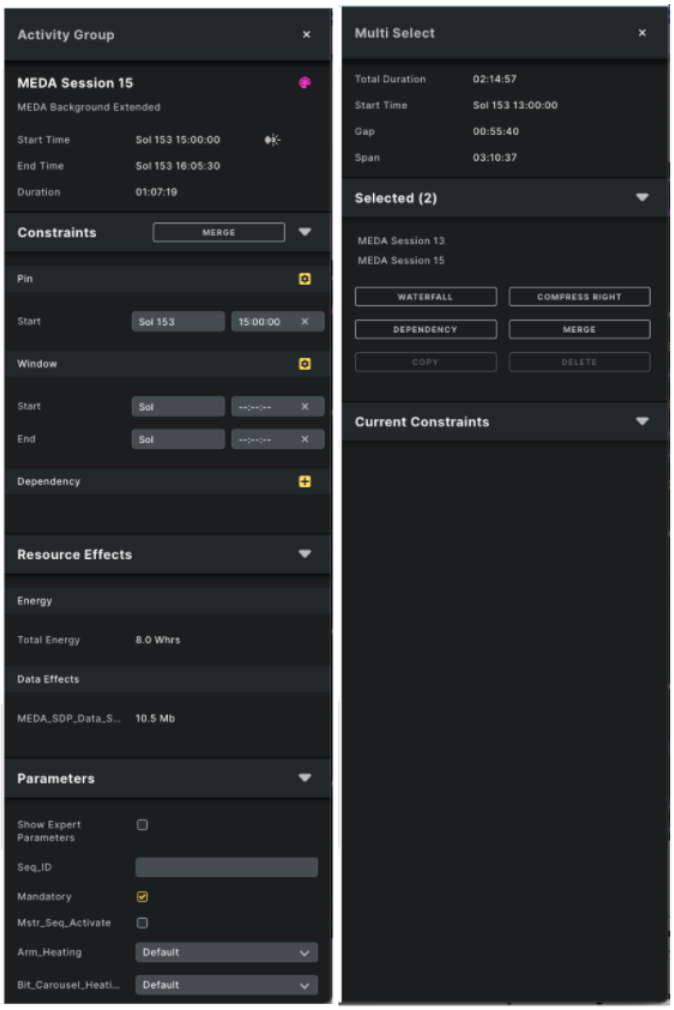


Figure 5. Detail Panel for a selected Activity Group (left). Multi-select Panel for multiple items selected (right).

COCPIT supports multiple selection of planning units, for which a Multi-Select Panel (Figure 5) opens on top of the Details Panel. This displays metrics based on the selection like Total Duration, Earliest Start Time, Span of the selection, as well as Gap or Overlap between two items. It also contains a set of operations to allow users to adjust relative timing, create Merge Sets, or delete multiple Components at once.

Timeline

The timeline is the main view where users review and manipulate the plan. It is a hierarchical timeline where the planning units are sorted by start time. At the top, sol number and hours are displayed with lines that span the height of the timeline, and each planning unit is rendered in a single row as a block in time order. On the left side, the row labels for each planning unit are displayed in a tree structure with additional icons. The icons allow for quick scan of important and relevant information about the planning unit for users. For example, if a merge set contains an activity that references a target, this information bubbles up to the top level to be easily

discoverable. A triangle is used to expand or collapse rows in the hierarchy to facilitate navigation along the timeline.

The plan has vertical blue dotted lines that identify handovers, which define the start and end boundaries of the planning sol. If more than one sol is being planned at once (either in a Look-Ahead Plan or a multi-sol Tactical plan), each planning sol will be separated by a handover as well. Handover times can be edited by the users.

When a Component is added to the plan, its Activity Groups appear on the timeline. The Activity Groups can be expanded to expose the Activities within. The order of the Activities within an Activity Group cannot be edited in the plan, but Activity Groups can be reordered and combined into Merge Sets. A Merge Set allows users to group Activity Groups from different Components together.

The highest level in the timeline is either Merge Sets, or Activity Groups that haven't been merged. These high-level planning units can be dragged across the timeline to change their start time, or can be manually updated in the Details Panel by changing the timing constraints.

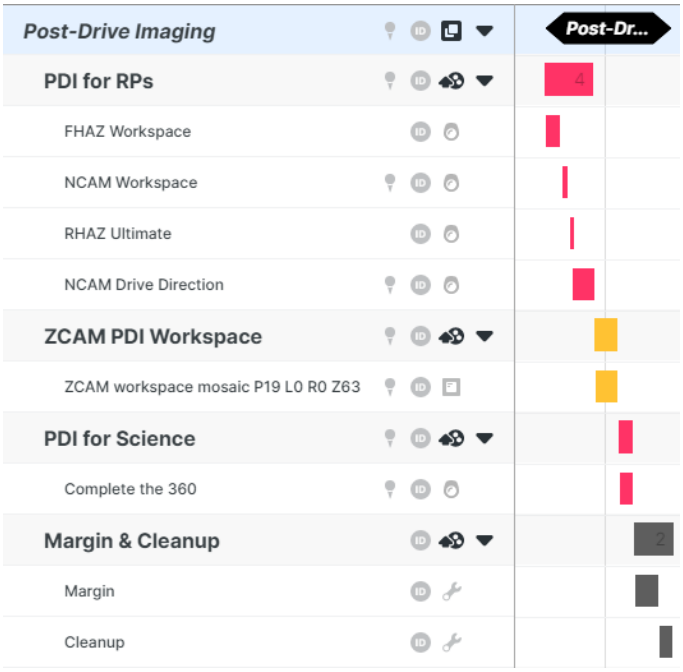


Figure 6. Merge Set structure in the Timeline

Activities and Activity Groups are rendered on the timeline as rectangles where the x-axis corresponds to time and width to duration. Merge Sets are rendered as blocks with diamond corners to differentiate them from Activity Groups. The background of the rows and row labels has various opacities of grey to aid in visually distinguishing the level of hierarchy (e.g. darker for higher levels). A light blue background on the row and row label indicates when a planning unit is selected.

In the example in Figure 6, see the Merge Set named Post-Drive Imaging. It consists of four Activity Groups (with gray row backgrounds), and each Activity Group contains one or more activities. Components are not shown directly on the timeline, but in this example the Activity Groups and Activities are color-coded based on subsystems (often spacecraft science instruments or collections of instruments), and users can see that 'PDI for RPs' and 'PDI for Science' Activity Groups are part of the same Component since they are regularly used together.

To assist in navigating the timeline, there are shortcuts to collapse or expand the hierarchy, scroll to a specific sol and zoom in and out. Additionally, a 'Focus Subsystem' feature allows a user to select a subsystem from a list, which causes the activities that belong to that subsystem to be expanded and highlighted, while the rest are collapsed.

As users move the cursor over the timeline, a vertical line always shows the Mars time under the mouse, and allows operators to add or move planning units to that specific time in the sol.

Activities can have an "overlay", a vertical shadow that spans the height of the timeline and the width of the activity duration. Overlays are color-coded, and assist in situational awareness when scrolling through the timeline. Overlays are commonly used for communication passes (pink and green overlays in Figure 7) or when the rover is asleep (grey overlay).

Merge Sets and Activity Groups can have earliest start and latest end time window constraints which are visualized on the timeline by a horizontal solid blue line at the top the row for that planning unit (See row MEDA Activate OT in Figure 7). Future versions of COCPIT will add other types of constraints visualization.

Overview

The Overview is an alternate view for the plan, in which users see the list of the Components in the plan sorted into columns based on sol (Figure 8). This view can be useful as a summary of the plan. Components that start before the first handover are shown in the Pre-Plan column. These may include uplink activities and activities that started but did not finish in the previous plan. Components that start after the last handover are shown in the Post-Plan column. Post-Plan items are generally for contingency planning, in case the next sol's plan doesn't reach the rover.

From the Overview, users can add and delete Components, move components between planning sols, and navigate the hierarchy of planning units. Detailed temporal information (i.e.

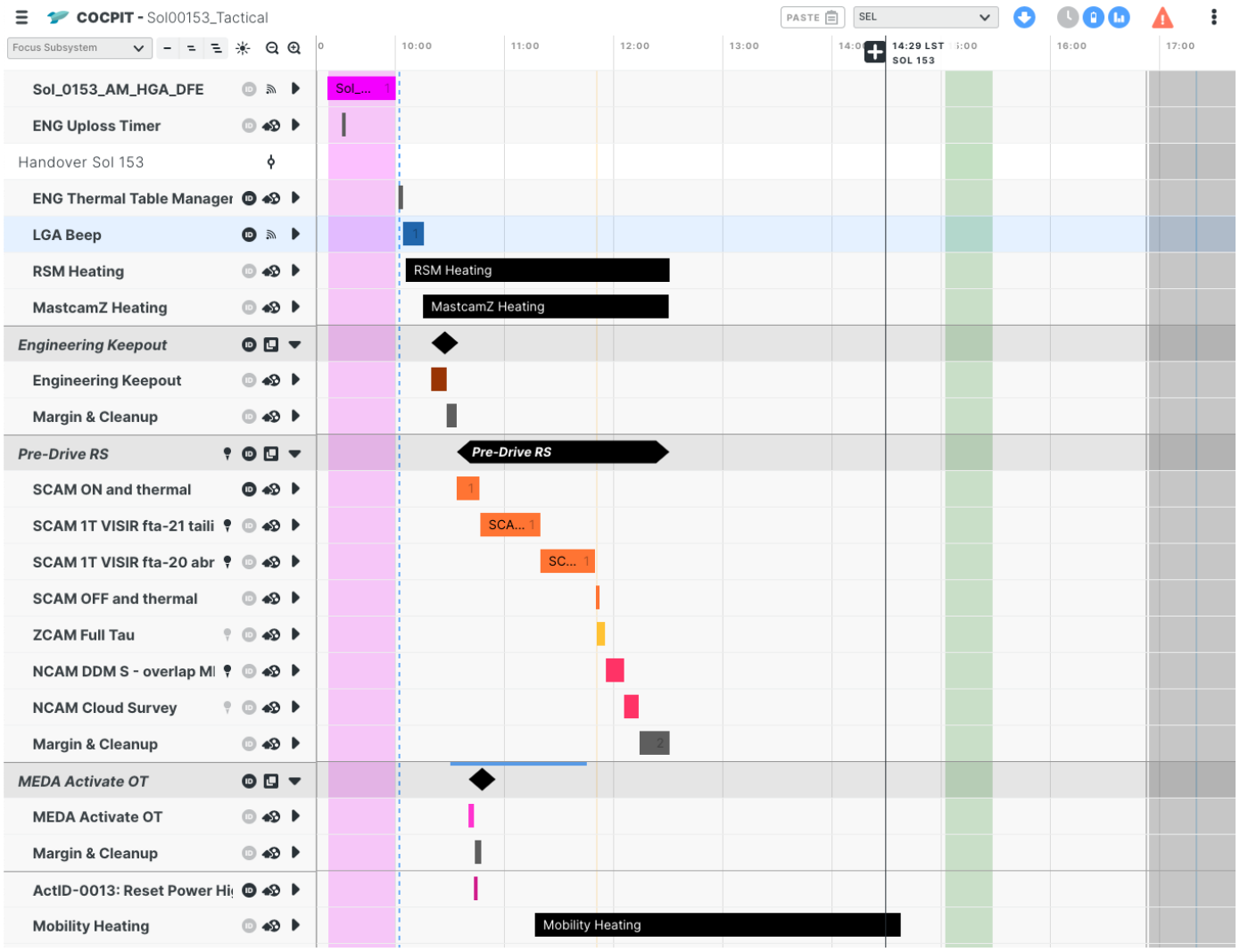


Figure 7. COCPIT Timeline

start, end and duration) is displayed in the Details Panel.

At the top of each sol column, users can set a default Science Intent Campaign. At the bottom of each column, there is a tray that displays data and power subtotals calculated for each planning sol. Resource goals per sol can be set here for long-term resource constraint planning.

5. Collaboration

One of the primary benefits of COCPIT compared to previous Mars planning tools is the real-time collaboration capability. Previous Mars surface missions heavily relied upon screen sharing, verbal requests to direct plan changes, copy and paste across plans, and merging changes from multiple copies of the plan together, which are more error prone and time consuming [1]. With COCPIT, team members around the world can access the plan remotely with a web browser. Members of the Engineering, Science, Instrument, and Robotic Operations teams all view and edit the plan together, instantaneously and automatically seeing each other's edits within the full context of the plan.

In COCPIT, collaborators can be team members using the web browser interface, or applications connecting to the APIs (see section 9). In either case, permissions are managed via roles. Roles in COCPIT are selected from a drop-down or included in the API calls, and enforcement is honor-based since many team members split their time among multiple roles. Each role has a specific set of permissions, dictating which COCPIT features they are allowed to use and which types of activities they are allowed to edit.

Power users own the highest-level planning units. They can add new Components to the plan, combine Activity Groups into Merge Sets, adjust constraints and timing, and apply other major changes to the plan. Roles that specialize in a specific subsystem have control of low-level activities for their instrument or subsystem by editing parameters, selecting targets, updating duration estimates, and adjusting data effects

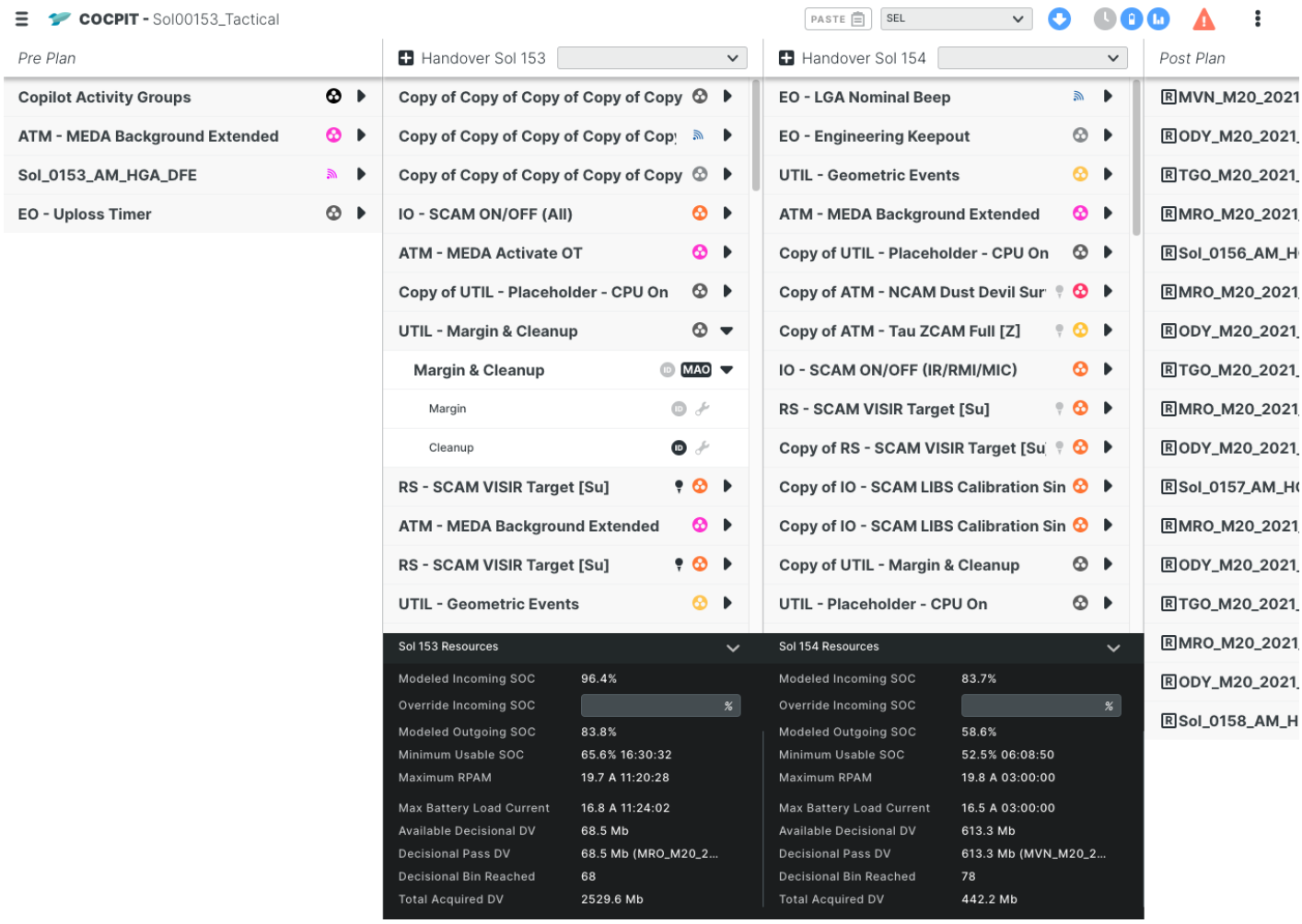


Figure 8. COCPIT Overview

directly. When a user edits a parameter, the new value is validated, ensuring that it is within the correct format or range as defined in the Activity Dictionary. Invalid values will show an error and not be saved to the plan, reducing risk for the planning process and the spacecraft.

During a Tactical Uplink shift, the first half is often characterized by members from each subsystem editing the plan in COCPIT simultaneously. When the changes have been finalized, the plan can be locked by a power user, preventing further edits. This gives the team confidence that the plan will not be changed inadvertently after their final review.

Permission Configuration

The COCPIT tool uses a JSON configuration file that defines the roles that tool operators can perform during the planning phases. This file also determines which COCPIT capabilities are allowed to be used by each role via the User Interface or APIs, defined by the mission management. A configuration file for roles allows changes in permissions without code changes. Additionally, a single, consistent roles and permissions schema can be shared with other GDS tools.

The configuration file allows inheritance of permissions from one role to another. Some permissions are statically defined, mapping to a specific action in the COCPIT User Interface. Others are dynamically determined based on the subsystem of the currently selected activity or the name of the specific parameter in question.

6. Resource Modeling

COCPIT uses formulas defined in the Activity Dictionary to calculate activity durations, simulate data generation, and predict power usage.

Duration

Duration is calculated on-the-fly from parameters that users can adjust during the planning process. For example, the duration of an imaging activity may vary depending on the selected image resolution and number of frames. When activities are grouped together serially in an Activity Group, any change in parameters that impact duration will automatically and immediately adjust start times of subsequent

activities. Some activity types allow overriding the duration to a specific parameter, instead of being derived from a duration formula.

Data Volume

COCPIT models the amount of data being generated, reprioritized, retransmitted, or downlinked by the rover. These are called data effects. Data effects are defined in the Activity Dictionary formulas, and are often parameterized so that the data volume and priority can be adjusted in the plan. Each activity may have multiple data effects, and each effect may add, remove, or manipulate data volumes on the Rover Compute Element (RCE) or in instrument memory (also called instrument data streams). Data on the RCE is prioritized for downlink by "bins" – from 1 (most critical) to 100. COCPIT models data resources by iterating through the activities in the plan chronologically, tracking the modeled data volume on the RCE, in each bin, and in each instrument data stream as it goes. When a UHF Pass is in the plan, COCPIT knows how much data volume can be downlinked by evaluating yet another Activity Dictionary formula. Nominally, COCPIT fills the downlink pass starting with bin 1 and continuing toward the lower priority data, until it has filled data volume of the pass. Subsequent activities may add additional data volume to the plan, and subsequent passes will downlink the remaining unsent data.

In some situations, it is ideal to downlink data out of priority order. For instance, if a specific orbiter relay has a long latency before the data will be received on Earth, the planning team may not want critical data included in that pass. The rover can be instructed to only include certain types of data on a high latency pass, and to accurately reflect that COCPIT can be instructed to adjust data modeling to limit which bins are considered for that pass.

This modeling is crucial to predicting when data acquired on Mars will be available for analysis on Earth. In each sol, a single UHF Pass is designated as the "decisional" pass, indicating that the data received on it is expected to have arrived on Earth before the next sol's Tactical Uplink process begins. As an example, Post-Drive Imaging is often considered critical data that must be downlinked on the decisional pass. After Perseverance drives, the Operations team needs images of the new location to begin discussing scientific targeting or new drive paths. COCPIT provides a warning when it predicts that this critical data will not fit within the decisional pass, as shown in Figure 9.

The data effects for each activity are displayed in its detail panel. COCPIT also allows viewing aggregated data modeling values at any time-point in the plan by right clicking on the timeline. A yellow dotted line on the timeline appears and a mid-plan resource tray opens, with the modeled data volume for every bin and instrument stream. This allows the team to see how data volumes change over time throughout the plan. If the team has set a limit for volume of data acquisition within a sol, COCPIT can also provide a warning when the limit is exceeded.

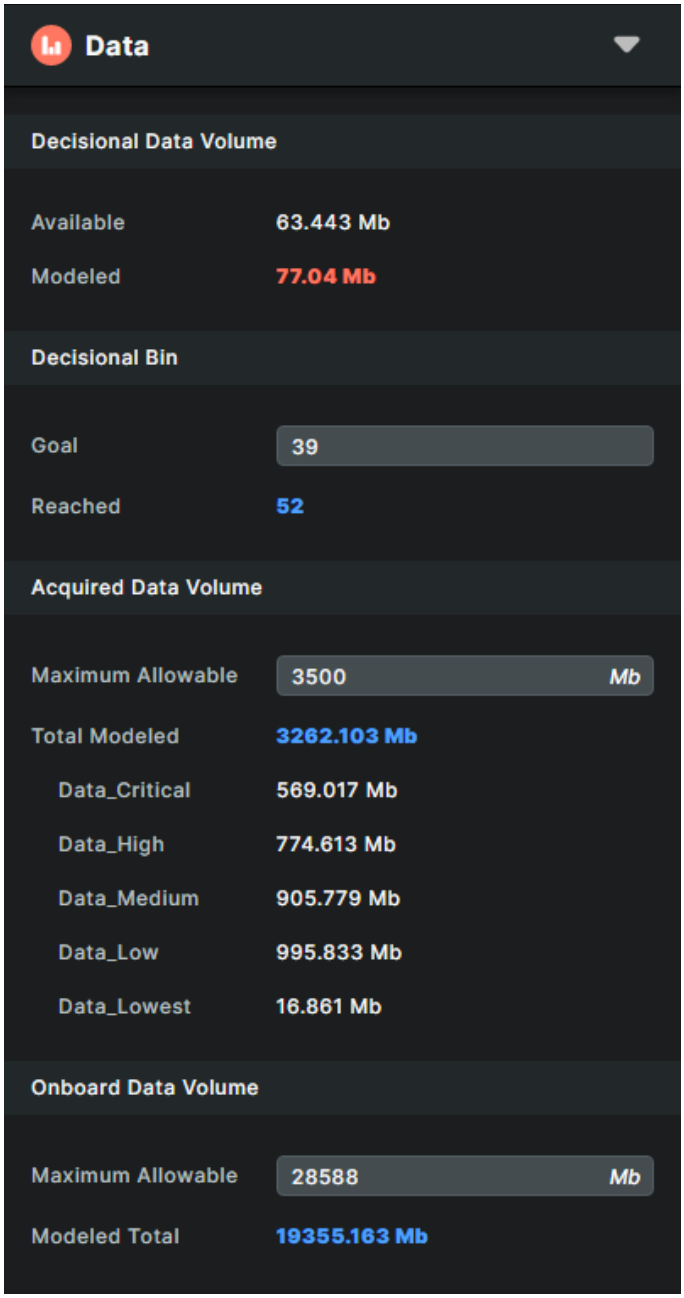


Figure 9: Data Modeling Summary

Perseverance often acquires more data than it can downlink in a single sol, therefore the modeled data volumes are propagated from one COCPIT plan to the next. Before being propagated, these predicted data volumes should be adjusted with real values to prevent modeling errors from being compounded across plans. Each Tactical Downlink shift receives telemetry with the latest actual onboard data volumes, which can be imported into COCPIT. However, by the time this data is received, the operations team is already planning

future sols. COCPIT must apply these known values at the appropriate time in a previous plan, then use those values as a starting point to model through any intermediate plans, and into the plan that the team is now working on. This provides the most up to date data modeling values so that the team can be confident in COCPIT's predictions.

Energy Usage

Another important resource for planning is energy usage. COCPIT calculates energy usage for each activity, including average and peak power. These calculations are based on Activity Dictionary formulas. Activities can be modeled with a constant rate throughout the duration, or with transitions that change the rate at calculated points within the activity. COCPIT can also detect when activities leave a power load on that may be turned off in a later activity. Power values are displayed in each activity's detail panel, and aggregated for higher level units like Activity Groups and Merge Sets.

COCPIT also integrates with the Multi-Mission Power Analysis Tool (MMPAT) to do more sophisticated power modeling. COCPIT calculates a set of power load changes based on activity times and durations. Using these, MMPAT calculates the initial state of charge, minimum state of charge, and final outgoing state of charge for the plan. It also determines the maximum electrical current, so that COCPIT can warn users if it is above the operational limit.

Like data volume, updated state of charge information is obtained from the rover in each Tactical Downlink shift. This updated information is passed through COCPIT to MMPAT to ensure modeling accuracy.

Power and energy modeling is imperative to ensure the safety of the rover, and to ensure that energy usage is budgeted appropriately so that future plans in the sol path can meet their objectives.

7. Temporal Constraint Network

COCPIT has two constraint modes which can be combined within a plan: pin-all and constraint-based planning. The beginning of the mission used pin-all planning, where each high-level planning unit is pinned to a precise Mars time. Users can drag and drop planning units on the timeline, waterfall or compress them against each other, or manually enter an exact time for the pin.

In constraint-based planning, high-level planning units can be given constraints to indicate the time ranges and circumstances in which they are allowed to be executed.

Dependency constraints indicate that a planning unit's execution is dependent on the execution status of another planning unit. For example, users may say that the Post-Drive Imaging block shall only happen after a drive has completed successfully, and must execute before the decisional comm pass occurs.

Time windows provide a range for the earliest start time and latest end time when a planning unit should execute on the rover, as seen in Figure 7. A preferred start time within the time window may also be requested but is not guaranteed. The time window can contain a mix of absolute Mars times and relative geometric constraints like sunrise, sunset, and solar noon. Extending the example above, Post-Drive Imaging may have a time window constraint to ensure it ends at least one hour before sunset, to ensure optimal lighting for imaging.

Pins, dependency constraints, and time windows can be used together within a plan to constrain each planning unit appropriately. COCPIT contains a temporal constraint network (TCN) to evaluate the set of constraints together and provide warnings when constraints are impossible to satisfy because they are inconsistent with each other or circular. By leveraging the temporal constraint network, COCPIT provides real-time feedback as planning units are moved and constraints edited, ensuring that all activities will fit within the plan. If, for example, the drive is extended such that the Post-Drive Imaging can no longer fit between it and the decisional pass, warnings will be displayed on the COCPIT timeline and in other areas of the user interface.

8. Autonomous Scheduling

It is important here to note the distinction between planning and scheduling. Planning requires deciding which activities are desired, and setting specific parameters and constraints for each of them. As noted previously, COCPIT is responsible for activity planning. Scheduling is the determination of an exact start time for each activity in the plan to execute. For this, COCPIT is closely integrated with an autonomous scheduling tool called Copilot [4].

For each Merge Set, Activity Group, and Activity in the plan, Copilot determines its scheduling restrictions. These include the predicted duration, timing and dependency constraints, acquired data volume, average and peak power, mechanical state requirements and effects, and allowable parallelism. Copilot also determines which activities need to be heated before they can be executed, and the required length of that preheat (depending on the season and time of sol). Similarly, it calculates when the CPU should be awake or asleep as appropriate to allow the rover to recharge between activities.

In the pin-all operational paradigm, Copilot's scheduling is highly constrained by the COCPIT plan. In this situation, Copilot validates that the explicit start times given on each planning unit do not violate any of the scheduling restrictions, and that valid heating and CPU activities can be created as necessary. If any of these conditions are not met, Copilot will not allow the planning unit to be scheduled. Constraint-based

planning would allow Copilot's algorithms to adjust the start times of activities to create a more optimal schedule.

In either paradigm, Copilot returns the schedule to COCPIT, where it is updated in the plan and sent to all clients. Users can ensure that everything was scheduled as expected, or can adjust the plan and constraints as desired to reach a satisfactory schedule.

The Perseverance rover's flight software is also capable of autonomous scheduling, using many of the same underlying algorithms as Copilot. When the Mars 2020 mission moves to using onboard scheduling, there will be no opportunity for human intervention in the final schedule results before they are executed. This makes it imperative to have a well-defined plan, and for COCPIT's planning interface to be intuitive and trustworthy to users.

9. Programmatic Interfaces

Application Programming Interface (API)

COCPIT does the heavy lifting of planning and scheduling on the web browser client with interactions on the Overview or Timeline View and Details Panel, but it lives in a software ecosystem with numerous tools across GDS. Therefore a set of APIs is provided to integrate with the tool. To respect the same permissions enforced in the COCPIT User Interface, a role name is passed as part of the data for plan-changing APIs, which confirm that the role is allowed to perform those actions.

External tools can submit changes to parameters and plan metadata. This endpoint checks permissions and validates the type, format, and range of the new value(s). In addition to individual parameter APIs, COCPIT also supports bulk adding and deleting of planning units by passing in a subset of the plan in the shared RML format. The COCPIT server reads the RML and makes the changes in the live plan.

COCPIT has APIs to inform other tools of important information like the id of the current Tactical Uplink plan or Look-Ahead Plan, and it allows downloading the plans as a whole or filtered by a set of subsystems depending on the user's needs.

The GDS team has developed a Helper Python Library that is released with every COCPIT version to be up to date with the plan-changing APIs as an intermediary to external Helper Scripts integrating with the COCPIT APIs.

Helper Scripts

COCPIT allows registering Helper Scripts, which then appear on a dropdown list in the user interface and are allowed to be run by certain roles. Helper Scripts are simply web services which take the COCPIT plan id as input, and use COCPIT APIs to read or modify the plan. The results from a Helper Script can be viewed in a COCPIT dialog or a new tab.

As an example, a critical Helper Script automates the creation of UHF and X band components in the plan based on predetermined details in an external database. In another case, Helper Scripts are frequently used to generate reports that summarize plan resource information.

Helper Scripts allow for rapidly adapting to process changes in operations since they can be updated independent of the COCPIT release cycle.

Subscription

COCPIT is responsible for the plan contents, but other GDS tools are interested in keeping track of changes happening to the plan. COCPIT has a subscription API where external tools can request notifications of changes filtered by a specific set of actions, planning unit types, or parameters. Each subscription has a filter and a destination URL to send the notifications to, and the subscriber will receive a message when the COCPIT server applies a change that matches the filter for that subscription until the subscription is canceled.

Plan Patch

While planning in COCPIT is mostly managed by modifying previously created Components, the Rover Planners (RPs) are not restricted in this manner for arm and mobility commanding. They need the ability to optimize the rover's operations by adding, removing and reordering activities. To accommodate this use case, COCPIT has a Plan Patch workflow.

COCPIT generates a snapshot of a subset of the plan, which the RPs import into their own tools to perform optimizations and modeling. The RP tools can generate an updated copy of the plan with their additions, deletions, and changes, which is uploaded back into COCPIT. This updated plan can be viewed by the rest of the Operations team. When approved, the differences can be automatically merged into the main Tactical plan, which has likely also been modified since the snapshot was taken.

This workflow is analogous to using the Linux tools diff and patch, although the underlying mechanisms are different. It could also be compared to creating a feature branch in a Version Control System like git, pushing separate changes to the feature branch and the main branch, and then merging the feature branch back into main. This is an iterative process that the Rover Planners can follow as many times as needed.

10. Sequencing

When the plan is finalized, its structure, parameters, and timeline can be used to generate the sequences that will run commands onboard the spacecraft. The separation between

high-level activity planning and low-level sequencing is common across JPL Mars missions [5]. The activity planning paradigm provides a high-level abstraction where the operations team can make meaningful assessments of the impact of different scheduling choices. However, the spacecraft is unable to read the activity plan, so we must create low-level sequences that pass explicit command directives to the rover. The structure of the plan is mapped to organizational sequences called the masters and submasters. The master sequence is the top-level sequence, and the design is such that there is exactly one master running at a time in most circumstances. The master invokes submasters and controls CPU wakeups and shutdowns at specific times as determined by the COCPIT plan. Submasters invoke individual sequences in parallel or sequentially, as defined by the structure of activities in the plan.

Often, the sequence call tree maps directly to activities in the plan. In some cases, the sequences themselves are automatically generated based on activity parameters in a process called Expansion. An expansion may take one of multiple forms – stored, template, scripted, and manual/tool-based. A stored expansion is a pre-built sequence used to implement an activity's intent. Template expansions allow for fill-in-the-blank replacement of sequence arguments based on activity parameters. Scripted expansions execute a script that takes the activity parameters as input and generates a sequence. Lastly, manual/tool-based sequencing is the fallback option when sequence creation requires inputs that are outside the scope of the activity plan, like Martian imagery or terrain files.

11. Summary

As missions become more complex, planning tools must allow for more collaboration, integration, and automation because longer planning cycles would mean more work for operators and less science done on the surface of Mars. COCPIT has proven to be an effective tool for coordination across a large remote team of operators, controlling powerful and highly configurable instruments and subsystems. COCPIT and Playbook are modern planning and scheduling tools paving the way for future software that support human and robotic exploration of the Moon, Mars, and beyond.

Acknowledgements

This work was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

We thank the COCPIT users who diligently assess the tool during and outside of operations and strategize the future requirements for continued improvement of Mars 2020 operations.

Thank you to the whole COCPIT management, design, test, infrastructure, and development team for working tirelessly to make this software successful. Anthony Robertson, Ryan Goetz, Kathryn Yu, Roxana Gonzalez Burgos, Joshua Camacho, Jimin Zheng, Sara Schnadt, Adrian Galvin, Basak Alper Ramaswamy, Guy Pyrzak, Hoan Luu, Natalie Rezek, and Usha Guduri.

References

[1] Pyrzak, G; Puncel, R; Vona, M; Lopez-Roig, R. "The Mars 2020 Ground Data System Architecture", in IEEE Conference on Aerospace, Big Sky, MT 2022.

[2] Marquez, J.J., Hillenius, S. Healy, M., and Silva-Martinez, J. (2019) "Lessons Learned from International Space Station Crew Autonomous Scheduling Test" International Workshop of Planning and Scheduling for Space (IWPSS), Berkeley, CA.

[3] Marquez, J.J., Hillenius, S., Deliz, I., Kanefsky, B., Zheng, J., and Reagan, M. (2017) "Increasing Crew Autonomy for Long Duration Exploration Missions: Self-Scheduling" in Aerospace Conference, 2017 IEEE, March 2017.

[4] Yelamanchili, A.; Agrawal, J.; Chien, S.; Biehl, J.; Connell, A.; Guduri, U.; Hazelrig, J.; Ip, I.; Maxwell, K.; Steadman, K.; and Towey, S. "Ground-based Automated Scheduling for the Mars 2020 Rover", in *Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation for Space*, Noordwijk, NL, 2020.

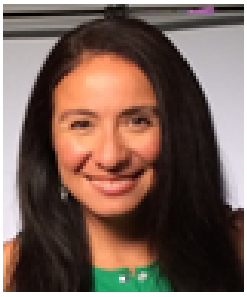
[5] Gildner, M.; Allbaugh, A.; Mishkin, A.; Algermissen, S.; Van Kirk, M.; Ellison, D.; Bridge, C.; Stough, T.; and Stroupe, A.. "Commanding Curiosity from the Couch: MSL Remote Operations, Challenges, and Path Ahead", in IEEE Conference on Aerospace, Big Sky, MT, 2021.

Biography

Ivy Deliz *received her B.S. in Computer Science from the Universidad de Puerto Rico - Rio Piedras and a M.S. in Computer Science from Columbia University in New York. She has been a software developer at the Human-Computer Interaction group at NASA Ames Research Center since 2010; currently under contract through ASRC Federal. She's the Co-Cognizant engineer for COCPIT. She's the Technical Lead for Playbook which supports several NASA Analog Missions including NEEMO and HERA to research future exploration concepts like scheduling crew autonomy. She has built planning tools for several NASA missions, including the International Space Station and Mars Science Laboratory.*

Andrea Connell *received a B.S. in Computer Science from the University of Wisconsin - La Crosse and a M.S. in Computer Science from the University of Hawai'i at Mānoa. She has been a software engineer at JPL since 2015, and is the Co-Cognizant Engineer for COCPIT and Copilot. She has also supported M20 Surface Operations as a Plan Integration Lead On Tactical (PILOT). Prior to joining Mars2020, Andrea worked for the Cassini Mission to Saturn and the Deep Space Network.*

Chet Joswig *received his B.S.E. in Computer Science and Engineering and MS in Computer Science from UCLA. He has been a software engineer at JPL since 2005. He has built planning and sequencing applications and subsystems for Mars Exploration Rovers, Mars Science Laboratory, and Mars2020 as well as R&TD operations tools. He supported MSL surface operations in several roles during early surface operations. On Mars 2020, he supported COCPIT development and was the Cognizant Engineer for Sequencer, the primary sequencing tool. He is currently the Technical Group Supervisor for Data Services.*



Jessica J. Marquez *received a B.S.E. in Mechanical Engineering from Princeton University, followed by a S.M. from the Department of Aeronautics and Astronautics at MIT. She received her Ph.D. in Human Systems Engineering from MIT. Since 2007, she has been working at the NASA Ames Research Center within the Human Systems Integration Division. As part of the Human Computer Interaction Group, she has supported the development and deployment of various planning and scheduling software tools for space missions, such as Playbook and SPIFe (Scheduling & Planning Interface For exploration) for the International Space Station and Mars surface missions. She is currently the SPIFe team lead.*

Bob Kanefsky *received his B.A. in Formal Systems from Stanford University. He has been a software developer at NASA Ames Research Center since 1988, currently in the Human-Computer Interaction group and employed by San José State University Research Foundation through a cooperative agreement with NASA. He is a COCPIT and Playbook developer and has written ground software for Mars Pathfinder, Mars Exploration Rovers, Mars Phoenix lander, Mars Science Laboratory, and ISS, and experimental flight software for New Millennium Deep Space 1, as well as software used in research.*